


**EE TIMES**  
 NETWORK

[LOGIN / REGISTER](#)
[SEAI](#)  
[Sear](#)  
[Con](#)  
[Sear](#)  
[LM3](#)
**SITE NAVIGATOR**
[Knowledge Centers ▸](#)
[Product Reviews](#)
[Data Sheets ▸](#)
[Guides & Experts ▸](#)
[News ▸](#)
[International ▸](#)
[Ask Us](#)
[Circuit Cellar](#)
[App Notes](#)
[NetSeminars](#)
[Careers](#)
[Resources ▸](#)
[FAQ](#)
[EE Times Network ▸](#)
[Electronics Group Sites ▸](#)

EE Expert Robert Ashby

## Embedded Engineering

[Archive](#) | [Main EE Expert](#) | [Guides & Experts](#)

### COM Port Programming

 by [Robert Ashby](#)

I wrote an [article](#) a while ago on communications via RS-232. It has been a simple method of serial communication that has had widespread application. Even with the popularity of the TCP/IP and the quick growth of USB, RS-232 still remains a simple quick link for devices without much overhead or excess hardware. I hope that by now you have tried a project using RS-232.

My article before showed you the mechanics of how the transfer occurs. I figured it would be good to give you a way to show off your RS-232 compatible device. COM port programming can be found in nearly every flavor of computer programming that I know about. However, you don't need to spend a lot of time or expense buying a programming software package (Though, you might want to hide this fact from your boss). You can build a neat little application that looks great quickly, and easily.

The secret is to use Internet Explorer. It's present on most Windows-based computers and allows you to present a plethora of neat visual content to the user with very little development. People get bored with the gray buttons and simple square boxes that make up a fair part of Windows usage. Internet Explorer will let you create the square boxes and simple buttons if that is your desire. It's also built to easily create colorful and interesting interfaces with a combination of pictures and media content.

Internet Explorer also will let you embed any Active-X component within a page. It shouldn't be a surprise that a Microsoft COMM control is a standard Active-X component available for use. The component needs to be properly registered on your machine for it to work correctly. You will also need the necessary .dll files required by the Active-X component to function properly. If you don't have everything set-up right, then your web page won't run the COM port as expected.

I won't spend any time talking about web page building. There are hundreds of sites on the web to help you out with that. Go to some that will give you a little help on javascript also. You can use VBscript to control your COM port object, but it will require a few more .dll files.

Add the following to the body of your page and you will have a COM port at your disposal.

```
<OBJECT classid=clsid:648A5600-2C6E-101B-82B6-000000000014 id=MSComm1>
```

**NetSeminar**  
 Services

```

<PARAM NAME="_ExtentX" VALUE="1005">
<PARAM NAME="_ExtentY" VALUE="1005">
<PARAM NAME="_Version" VALUE="393216">
<PARAM NAME="CommPort" VALUE="1">
<PARAM NAME="DTREnable" VALUE="1">
<PARAM NAME="Handshaking" VALUE="0">
<PARAM NAME="InBufferSize" VALUE="1024">
<PARAM NAME="InputLen" VALUE="0">
<PARAM NAME="NullDiscard" VALUE="0">
<PARAM NAME="OutBufferSize" VALUE="512">
<PARAM NAME="ParityReplace" VALUE="63">
<PARAM NAME="RThreshold" VALUE="0">
<PARAM NAME="RTSEnable" VALUE="0">
<PARAM NAME="BaudRate" VALUE="9600">
<PARAM NAME="ParitySetting" VALUE="0">
<PARAM NAME="DataBits" VALUE="8">
<PARAM NAME="StopBits" VALUE="0">
<PARAM NAME="SThreshold" VALUE="0">
<PARAM NAME="EOFEEnable" VALUE="0">
<PARAM NAME="InputMode" VALUE="0">
</OBJECT>

```

Most of the parameters should be self-explanatory. I would recommend that you review the datasheet for the 16550 uart chip for a refresher if needed at [National's website](#). It would also pay to search around Microsoft's [website](#) for information on the Microsoft Communications Control.

Javascript will give you an easy way to control the characteristics of the COM port control giving you quick access to the methods and properties of the COM port, e.g. `MSComm1.PortOpen = true`. Input and output of the COM port is handled as strings.

Here are a few little interesting tidbits of information that I would have like handed to me in the beginning.

The default mode of the COM control is text mode. That means that the control will add on return and line feed bytes that your embedded system might not be expecting. The extra characters can be avoided by setting the `InputMode` property to "1" which is binary mode. You should also be aware that `NullDiscard` will prevent the reception of null characters if set to "true."

If you need to send nonprintable characters to the output, you can use the escape character, i.e. `"\x41"` is the same as "A." This will allow you to specify any character from 0x00 to 0xFF. The input might come back as nonprintable characters also. The easiest way that I have found to separate the value of the characters is to use the method `CharCodeAt(n)`. It returns the unicode value of character at index 'n' within a string.

The best teacher has always been example, so I have included an example of making a COM Port work. Make sure that the Active-X object is properly registered and that the security settings in Internet Explorer are set to allow Active-X objects to run.

```

<html>
<head>
<title>COM Port Communications</title>
<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
<!--
function ComPortButton_onclick(){
if (MSComm1.PortOpen==false){
    MSComm1.PortOpen=true;
    ComPortButton.value="Close";
}
else{
    MSComm1.PortOpen=false;
    ComPortButton.value="Open";
}
}

```

```

    }
}
function SendButton_onclick() {
if(MSComm1.PortOpen==true){
    MSComm1.Output="Here is an ASCII character"+"\\x41";
}
else{
    alert("You need to open the COM port first");
}
}
function MSComm1_OnComm() {
var InputString;
var Number;
InputString=MSComm1.Input;
number = InputString.charCodeAt(0);
alert(Number);
}
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=MSComm1 EVENT=OnComm>
<!--
MSComm1_OnComm()
//-->
</SCRIPT>
</head>
<body>
<P>This web page communicates over a COM port.</P>
<P>
<OBJECT classid=clsid:648A5600-2C6E-101B-82B6-000000000014
id=MSComm1>
<PARAM NAME="_ExtentX" VALUE="1005">
<PARAM NAME="_ExtentY" VALUE="1005">
<PARAM NAME="_Version" VALUE="393216">
<PARAM NAME="CommPort" VALUE="1">
<PARAM NAME="DTREnable" VALUE="-1">
<PARAM NAME="Handshaking" VALUE="0">
<PARAM NAME="InBufferSize" VALUE="1024">
<PARAM NAME="InputLen" VALUE="0">
<PARAM NAME="NullDiscard" VALUE="0">
<PARAM NAME="OutBufferSize" VALUE="512">
<PARAM NAME="ParityReplace" VALUE="63">
<PARAM NAME="RThreshold" VALUE="0">
<PARAM NAME="RTSEnable" VALUE="0">
<PARAM NAME="BaudRate" VALUE="9600">
<PARAM NAME="ParitySetting" VALUE="0">
<PARAM NAME="DataBits" VALUE="8">
<PARAM NAME="StopBits" VALUE="0">
<PARAM NAME="SThreshold" VALUE="0">
<PARAM NAME="EOFEEnable" VALUE="0">
<PARAM NAME="InputMode" VALUE="0">
</OBJECT>
</P>
<P>
<INPUT id=ComPortButton name=ComPortButton type=button value=Open
LANGUAGE=javascript onclick="return ComPortButton_onclick()" style="LEFT:
340px; TOP: 33px"> <INPUT id=SendButton lowsrc="" name=SendButton style="LEFT:
68px; TOP: 110px" type=button value="Send Message" LANGUAGE=javascript
onclick="return SendButton_onclick()"> </P>
</body>
</html>

```

It's really not that difficult even for the meekest of programmers. It's easy to add to the page to get some flashy content that will really add some shine to your project. It is even fast enough that you could make a side derivation for the kids at home or even your significant other if it's someone a lot more "techy" than my wife.

### Embedded Engineering

[Guides and Experts](#) [Analog Avenue](#) [EDA Tools](#) [PLD](#) [DSP](#) [EDA](#) [Embedded Systems](#) [Power](#) [Test](#)

**Electronics Marketplace**

[Click here to get your listing up.](#)


**EETIMES**  
 NETWORK

[LOGIN / REGISTER](#)

 SEAI  
 Sear  
 Con  
 Sear  
 LM3

**SITE NAVIGATOR**
[Knowledge Centers ▶](#)
[Product Reviews](#)
[Data Sheets ▶](#)
[Guides & Experts ▶](#)
[News ▶](#)
[International ▶](#)
[Ask Us](#)
[Circuit Cellar](#)
[App Notes](#)
[NetSeminars](#)
[Careers](#)
[Resources ▶](#)
[FAQ](#)
[EE Times Network ▶](#)
[Electronics Group Sites ▶](#)

EE Expert Robert Ashby

# Embedded Engineering

[Archive](#) | [Main EE Expert](#) | [Guides & Experts](#)

## How to Communicate Via a COM Port with Web Pages

 by [Robert Ashby](#)

I was asked some time ago by a reader to give a more detailed example of how to use COM ports in web pages. He had read my [article](#) on how to communicate via a COM port with web pages. His task was the testing of modems to see whether they could send and receive information and how to log the data response.

Javascript and ActiveX objects are easy to add to your web page and don't require any tools fancier than a simple text editor. However, the use of these tools can be greatly facilitated with additional tools such as the Microsoft Development Environment that comes with Visual Studio. Visual J++ is another great tool. If you are cheap (like me) then you might just try your hand first with 'Notepad' or any other favorite text editor.

Not an authority on modems, I created a small [page](#) that would query the modem attached to the COM port for some simple information and log that information to a file. Copy this file onto your computer to try it out.

**NetSeminar**  
 Services

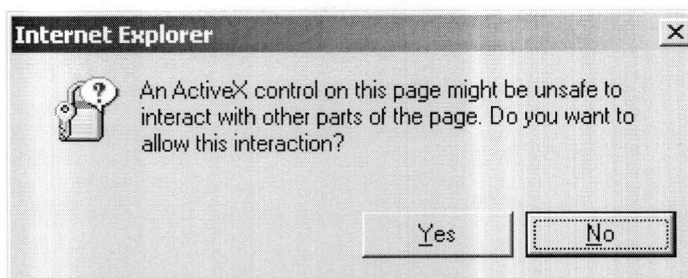
**WARNING** this page creates the file C:\testfile.txt on your hard drive. In the unlikely event that you already have a file by this name on your hard drive, it will be overwritten. Please either back up that file or make sure you edit this provided file so you don't overwrite your information.

I have only tested this page with computers running Windows 2000 and 98 and Internet Explorer 5.01. If the program isn't working for you, check to make sure that you have the MSComm ActiveX object registered on your computer. If it's still not working and you're ready to pull your hair out, drop me an e-mail and I'll try to help.

The property about the MSComm ActiveX object that seems to trip me up most often is the Input Mode property. If it is set to 0 (text) then the information is retrieved from the port as a string. If it is set to 1 (binary data) then the information is structured as a byte array. If you are just dealing with ASCII characters and don't care much about possible appending returns and line feeds, then leave it at 0.

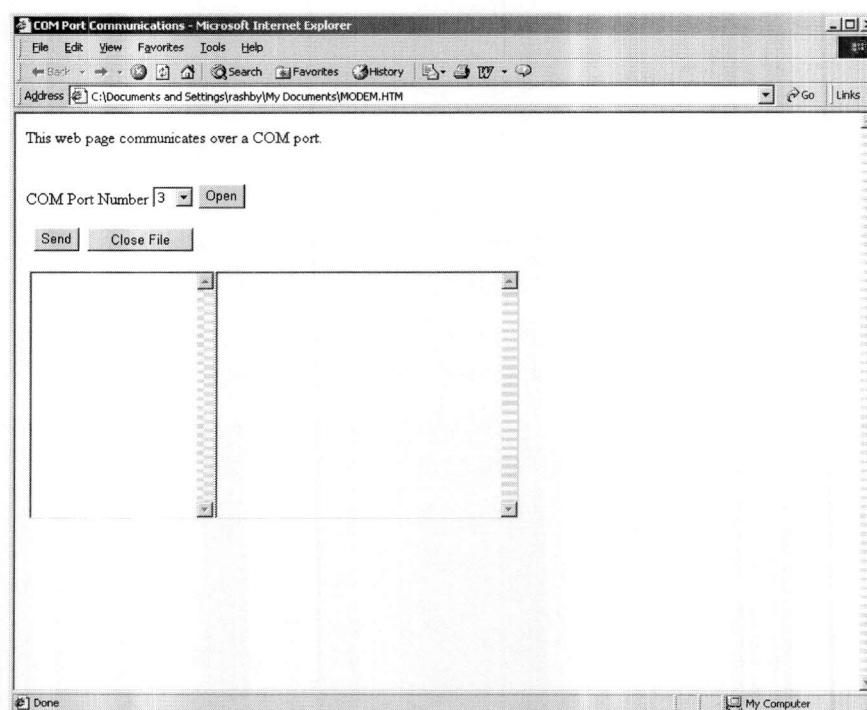
I've left the file simple so you don't have to wade through a bunch of bells and whistles to figure out what is going on.

When the file is first loaded you will probably get a prompt like this:



Select yes or the web page won't work. You are getting this prompt because the ActiveX MSComm object is embedded in the page. When the file opens, it will create the file "C:\testfile.txt" on your hard-drive. Any text that is received from the COM Port while this page is open and the file has not been closed will be recorded into this file. The file object is set up to overwrite any existing files.

Here is a quick shot of what the page looks like:



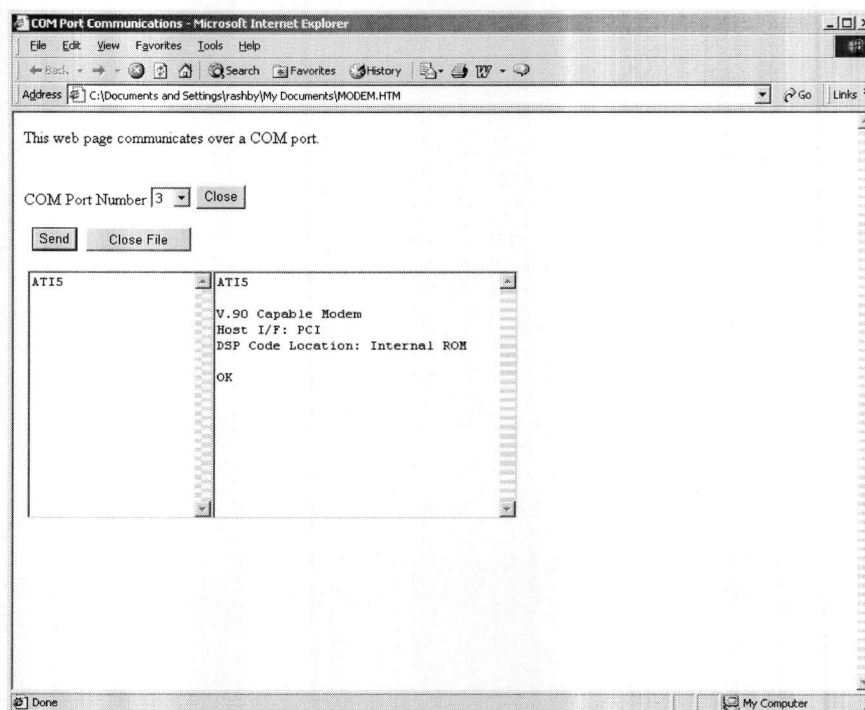
The COM Port Number should be set to the port that your modem resides on. Or if you are using this file to check your own design, then set it to the appropriate port number. If you are using ports greater than 4, then you can edit that object on lines 71 — 74 to match the desired port number.

Once you have opened the port you can type a line in the left-hand text area. This is the SendWindow. When you press the SendButton, the text in the SendWindow is sent to your respective COM Port. I have set the COM Port up to be 9600 baud, 8 bits of data, 1 stop bit, no parity.

Any text that is received from the COM Port is echoed in the right-hand text area (ReceiveWindow). This information is also written to the file that was opened when the page was first initialized. If you want to stop recording to the file, you can press the Close File button, which will close the file and prevent any further

writing to that file. If you refresh the page, then the file is reinitialized and the old information is gone. You can edit the file as an ordinary text document.

If you are sending an AT type command to your modem be sure to follow the command with a return so that the command is recognized by the modem. If you want to send several commands in series, you will most likely need to add some timing to your web page to allow the modem time to respond to your first command before proceeding to any additional commands. Here is an example of what my computer looks like after sending the command ATi5:



The modem's response has been logged to a file on my hard drive. Pretty cool! There's not that much to it, but it can be a very powerful tool.

E-mail me if you would like some more in-depth articles in this area.

## **Embedded Engineering Archive**

[Guides and Experts](#) [Analog Avenue](#) [EDA Tools](#) [PLD](#) [DSP](#) [EDA](#) [Embedded Systems](#) [Power](#) [Test](#)

**Electronics Marketplace**

[Click here to get your listing up.](#)

Copyright © 2003 ChipCenter-QuestLink

[About ChipCenter-Questlink](#) [Contact Us](#) [Privacy Statement](#) [Advertising Information](#) [FAQ](#)